REPORT NO. FR-87-72-0042

PERFORMANCE MODELLING OF AUTONOMOUS ELECTRO-OPTICAL SENSORS

Third Semi-Annual Technical Report
Contract No. DAAB10-86-C-0534

M.A. Lepley, W.G. Hanley, R. Chellappa

November 1987

REPORT PERIOD
15 May 1987 through 15 November 1987

Prepared For:

U.S. ARMY CENTER FOR
NIGHT VISION AND ELECTRO-OPTICS
FORT BELVOIR, VIRGINIA 22060

Image and Signal Processing Laboratory
Tactical Engineering Division
Hughes Aircraft Company • El Segundo, California 90245

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION: Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S): N/A (Ref) FR-87-72-0042	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION: Image & Signal Processing Laboratory, Tactical Engineering Division; 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION: U.S. Army Center for Night Vision and Electro-Optics
6c. ADDRESS (City, State, and ZIP Code): Hughes Aircraft Company, El Segundo, Calif. 90245	7b. ADDRESS (City, State, and ZIP Code): Fort Belvoir, Virginia 22060
8a. NAME OF FUNDING/SPONSORING ORGANIZATION; 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER: DAAB10-86-C-0534
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS: PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO

11. TITLE (Include Security Classification)

Performance Modelling of Autonomous Electro-Optical Sensors — Third Semi-Annual Technical Report

12. PERSONAL AUTHOR(S)

Lepley, Margaret A; Hanley, William G; and Chellappa, Rama (Univ. of S. Calif.)

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Semi-Annual Progress	FROM 5/87 TO 11/87	1987, November 15	31

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	laser radar target recognition, robust surface estimation, M-estimates, segmentation, crease edge detection, filters for edge detection

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

In this report, our investigation of segmentation of LADAR imagery is documented. The research, implementation, and testing of several aspects of segmentation are all addressed. The topics covered include:

- the robustness of various surface estimation techniques,
- removal of the majority of image background,
- region growing technique for separating object surfaces,
- smoothing filters for improving derivative related computations,
- crease edge detection.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
☑ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL: Vince Mirelli	22b. TELEPHONE (Include Area Code): (703) 664-5635; 22c. OFFICE SYMBOL

DD Form 1473, JUN 86 — Previous editions are obsolete. — SECURITY CLASSIFICATION OF THIS PAGE: UNCLASSIFIED

Table of Contents

List of Figures

List of Tables

1.0 INTRODUCTION

The Performance Modeling of Autonomous Electro-Optical Sensors (PM) program is a basic research effort aimed at the fundamental analysis of image processing methodology. To this end, an investigation into the the use of LADAR imagery for target recognition was initiated. Preliminary research and methodology development occurred from November 1986 to May 1987. Research and implementation are continuing and this report details the investigation performed and results obtained during the period from May 15, 1987 through November 15, 1987.

During the previous six months, the LADAR target recognition investigation has focused on the area of object segmentation. The issues that were addressed include:

- Surface segmentation via region growing,
- Robust planar surface estimation,
- Background plane removal,
- Smoothing of the image prior to the estimation of derivatives,
- Crease edge detection.

The introduction to Section 2.0 contains a general problem definition and a brief methodology overview. The various subsections describe those aspects of segmentation that were addressed in this reporting period. Specific techniques being investigated are detailed, along with any pertinent demonstrations that have been performed. More information on the overall recognition methodology and matching in particular are contained in the Second Semi-Annual Report [7]. Areas of future work and investigation are summarized in Section 3.0.

2.0 TECHNICAL PROGRESS

In the Second Semi-Annual Report [7] we described our approach to target recognition based upon laser radar sensor information. This approach emphasizes (i) the 3-dimensional geometry of the targets and the scene, (ii) both local and global features, and (iii) model-based graph matching. In [7] an overview of segmentation, description, and matching were given, and detection was touched upon briefly. The major focus of investigation during that period involved matching, with a proof of concept example demonstrating the feasibility of the selected approach. During the last six months, the focus has shifted to the development of an effective segmentation methodology.

This report contains a detailed discussion of the areas that were investigated during the segmentation study. Overviews of the algorithms that are being researched are presented along with examples of the associated experimental results.

2.1 Segmentation/Description Overview

With range imagery the segmentation step involves determining the extended structure of an object within the given target window. Specifically, this procedure entails: (i) the detection of target boundaries, (ii) the location of edges between surfaces of the target, and (iii) the determination of target surfaces, but not necessarily in that order.

Segmentation of intensity images is a well studied problem in image understanding and computer vision applications. On the contrary, segmentation of range images for target/background separation, and for detecting edges between surfaces is a relatively new research topic.

There are two distinct classes of methods for detecting edges. The first is to search for edges *directly* by hunting for properties that distinguish an edge. There are many different edge detectors of this type, some of which are more robust to noise than others. Another method for finding edges is *indirect*. First, the surfaces of the object are identified and then the intersection of two adjacent surfaces indicates the presence of an edge. Through the combination of these two approaches, more accurate estimates of both surface and edge information can be obtained.

An iterative region growing technique for identifying surfaces also provides a "parametric" representation of the surfaces. Such a representation is critical for effective object description. Specifically, surface parameterization is required to reduce the description of a geometric entity to a small set of measures which are invariant to rotations, translations, and changes in scale. In other words, it is the first step in forming a general object description that can be employed for matching. Therefore by using a surface-based segmentation in conjunction with edge detection the functions of segmentation and initial description can be accomplished simultaneously with a high degree of accuracy.

Besl and Jain [1] have developed a fast region growing segmentation algorithm which starts from seed regions based on uniform invariant curvature classification and proceeds by parallel, iterative region growing. We are using this surface-based algorithm as the core of our segmentation approach. Section 2.2 describes this technique along with our modifications, and presents some examples of its behavior.

One of the most important components of the region growing algorithm is the surface estimation which is effectively reduced to planar estimation. During preliminary tests of the matching technique in February 1987 [4], it was observed that planar estimation of the noisy range data was extremely erratic when a least square fit was used. Therefore a comparison study of various planar estimation techniques was performed. Section 2.3 describes the various estimation methods that were compared as well as the tests that were performed and their results.

Another issue which arose was the appropriate estimation of thresholds used during the region growing process. Many of the thresholds are computed from the image, but may be set incorrectly if too much background is included in this computation. This occurs because the background pixels often have a larger variance from a planar fit since (i) they generally don't come from exact planes and (ii) a high angle of incidence makes the footprint of the pixel cover a wide interval of ranges. Ideally the thresholds should be calculated only from pixels on the target itself. Using LADAR imagery alone though it is not easy to extract the target boundary. Section 2.4 discusses the problems associated with finding and eliminating a large percentage of background pixels during pre-processing for the region grower.

During the region growing, and for most types of edge detection, it is necessary to estimate various partial derivatives from the image. Due to noise and quantization error in the image, these estimates are often unreliable. Smoothing can be used to improve derivative estimation if it is performed appropriately. Section 2.5 discusses how regularization can be employed to optimally filter an image for a specified operation.

The two types of edges that occur in range imagery are (i) *jump* edges which are formed when there is a discontinuity (other than a range ambiguity) in the relative range, and (ii) *crease* edges, which are derived from discontinuities in orientation without discontinuities in range. Jump edges can be identified quite accurately either directly, or indirectly using the region growing algorithm. Crease edges can be more difficult to identify via region growing and their location may vary. Therefore an alternate crease edge detection technique has been implemented to supplement and correct the output of the surface-based segmentation. The crease edge detection technique used is a variant of that proposed by Mitiche and Aggarwal in [8]. This technique is described in section 2.6 and an example is presented demonstrating its function.

2.2 Surface-Based Segmentation

The goal of surface-based segmentation is to accurately estimate the true underlying surface shape of objects in LADAR range imagery. The technique developed by Besl and Jain [1] employs iterative region growing, and is based upon two significant assumptions regarding the characteristics of targets in range data.

- All objects of interest may be represented by piecewise smooth surfaces.
- Planar and/or quadratic surfaces are sufficient to model targets under typical conditions of sensor resolution.

Seed regions form the starting point for the region growing process. They should represent small, homogeneous regions not containing any boundary points and be a reliable representative of the surrounding surface curvature. Besl and Jain use the Gaussian (K) and mean (H) curvature to form a HK-Sign map which divides the image into regions belonging to an invariant curvature class. The eight possible curvature classes – peak, pit, ridge, valley, flat, minimal, saddle valley, and saddle ridge – are used as a starting point when forming seed regions. Gaussian and mean curvature are defined and discussed in more detail in [1,7].

A summary of the major steps involved in this algorithm is presented here.

1. Divide the image into regions of invariant surface class. This is done using the HK-Sign map.
2. Take the largest connected region of invariant surface class and contract it so that only points interior to the region remain. This is the seed region.
3. Fit a plane (or higher order surface) to the seed region. This is the estimated surface.
4. Search for all the points in the image that are approximately fit by the estimated surface and find the largest connected subset which overlaps the seed region. This becomes the new seed region.
5. Check conditions for stopping. If region growing is to continue go back to step 3. There are a variety of conditions related to stopping at this point. Briefly, the more important checks involve (i) the error in the surface fit vs. a calculated threshold, (ii) the relative change in region size since the previous iteration, and (iii) the current number of iterations.
6. If the final region is acceptable then add it to the surface list and remove the points in that region from the surface class image. Otherwise, remove the original seed region points from the surface class image. This is done to prevent looping caused by repeatedly trying to fit the same surfaces. The acceptance of a region is also based upon a variety of conditions, related largely to region size, error of fit, and whether this region is contained within a previously accepted region.
7. Go back to choose another surface class region in step 2.

The purpose of this investigation was to determine whether this technique could be effectively used for segmenting surfaces of vehicular targets. To address this question several alterations were made to Besl's algorithm.

- Initial runs with the algorithm showed a tendency to use a parabolic fit over the entire target. Since the target types of interest are composed of planar surfaces, the surface fitting was restricted to planes. This restriction is being used for these initial tests and may be relaxed later.

- The planar surface fitting is being performed using an estimation technique that our analysis showed to be superior to least squares under contaminated noisy conditions. This technique is called M-estimates. It is presented in section 2.3 along with the tests comparing it to alternative estimation methods.

- With the addition of M-estimates for plane fitting, a portion of the code which attempted to remove outliers in a very adhoc manner was removed. M-estimates down-weights the real outliers much more precisely.

- The variance of the gradient in the background was often higher than on the target itself, and many of the thresholds are set based upon the perceived gradient variance in the entire image. Therefore, the thresholds were being skewed due to background noise and the segmentation of subregions within a target was poor or non-existent. By masking out a sizable fraction of the background pixels when calculating thresholds the performance was much improved. Since surface segmentation generally occurs after the target is mostly separated from the background, a mask is very appropriate. Section 2.4 discusses how the background masks can be obtained.

- The region growing algorithm does not take into consideration any explicit edge information. Jump edges are usually not a problem since a region will not easily grow beyond an existing jump edge. On the other hand, crease edges are not easily accounted for since surface fitting alone may grow beyond simple discontinuities in orientation. Therefore code has been added that can be used to limit region growth based upon a mask. Such a mask would be made based upon crease and jump edges in the image. This option has been tested and works functionally. It could prove very effective when integrating other types of sensor information. For example, edge information provided by a passive IR image could be taken into account.

So far we have seen that the target regions can be segmented into separate planar surface regions, and the reconstruction image visually resembles the original image with much less noise. But we have found that oversegmentation often occurs, with too many subregions being formed. Post processing to merge similar regions that are adjacent would help eliminate this problem. Crease edge detection will be used at this point to help determine if two regions are distinct or if they need to be merged.

2.3 Robust Planar Surface Estimation

This investigation was initiated when it was observed that the planar fits that were being found for sections of range imagery were highly erratic. Error in surface estimates may be due to several different causes.

- Expected variance.
- Incorrect modelling of the noise process.
- Data samples that do not come from a single underlying distribution.

- Invalid parametric model.

It is always desirable to chose a technique which has low expected variance but other points must be considered as well. An estimation technique which works well under specific conditions but degrades rapidly with very minor changes in the underlying distribution is not useful when there is some doubt as to the form of the noise process. Parametric surface estimation techniques generally assume all the data points belong to the same underlying distribution. For instance if gaussian noise is the only problem, then the generally accepted estimation technique is least squares regression. When *corrupting data* which belongs to a different underlying distribution is present then the original surface estimation technique will have degraded performance. A problem also arises when it is suspected that the parametric model is inaccurate. For instance, does all the data really come from one plane, or does part of it belong to another surface. Some estimation techniques can handle the addition of corrupting data more readily than others. Since some LADAR returns may be corrupted and slight missegmentations may also occur, it is important to find a technique that is robust under these conditions.

In this section we discuss briefly the most promising techniques for planar estimation in the presence of considerable noise and some corrupting data. Comparisons of their behavior on controlled data are also detailed. These methods include: least squares regression, M-estimate regression, and the removal of outliers based upon a weighting function. Each of these techniques is described and compared in the following sections.

Techniques

Initially the data is assumed to come from a plane with additive noise. The position variable is $\mathbf{x}_i = (1, x_i, y_i)$, and the corresponding range observation is z_i. Then we say

$$z_i = \beta_0^0 + x_i\beta_1^0 + y_i\beta_2^0 + e_i = \mathbf{x}_i\beta^0 + e_i$$

where the e_i are independent, identically distributed noise processes with mean 0 and standard deviation σ, and $\beta^0 = (\beta_0^0, \beta_1^0, \beta_2^0)$ are the planar coefficents. If there are n data points then we have

$$\mathbf{z} = X\beta^0 + \mathbf{e}$$

where X is the matrix of $\mathbf{x}_i$.

Least Squares: The goal in least squares is to estimate β so as to minimize the difference between the predicted range values, $\mathbf{x}_i\beta$, and the observed values, z_i, i.e.

$$\min_{\beta} \sum_{i=1}^{n} (z_i - \mathbf{x}_i\beta)^2$$

This minimum occurs when

$$\hat{\beta} = (X^T X)^{-1} X^T \mathbf{z}$$

As one would hope the expected value of $\hat{\beta}$ is β^0. The covariance of $\hat{\beta}$ is

$$\text{Cov}(\hat{\beta}) = \sigma^2(X^T X)^{-1}.$$

Thus, if the region to be fit is a square of area n centered at the origin, then

$$\begin{aligned} \text{Var}(\hat{\beta}_0) &= \sigma^2/n \\ \text{Var}(\hat{\beta}_1) &\approx \sigma^2(\frac{12}{n}) \\ \text{Var}(\hat{\beta}_2) &\approx \sigma^2(\frac{12}{n}) \end{aligned}$$

So if σ is large, as is often the case in range imagery, then a large number of pixels is needed in each region to obtain an accurate fit. The variance of $\hat{\beta}_0$ will depend upon the location of the origin relative to the center of the region being fit, with a larger distance inducing a larger variance. The slope variances do not change with translation, but they will change as the shape of the region is altered. For example, a region which is long in the x direction but narrow in the y direction will have a small variance for $\hat{\beta}_1$ but a larger variance for $\hat{\beta}_2$. A more compact region that has the same area would have a larger variance for $\hat{\beta}_1$ and a smaller variance for $\hat{\beta}_2$.

However, sometimes the observed data does not exhibit the aforementioned behavior. In other words the above description of e_i is incorrect. Specifically, certain data points may be corrupted by more than the expected noise. Since least squares tries to minimize the squared error, undue weight may be given to points that are out of line with the rest of the data. For example, consider the least squares regressions displayed in Figure 1.

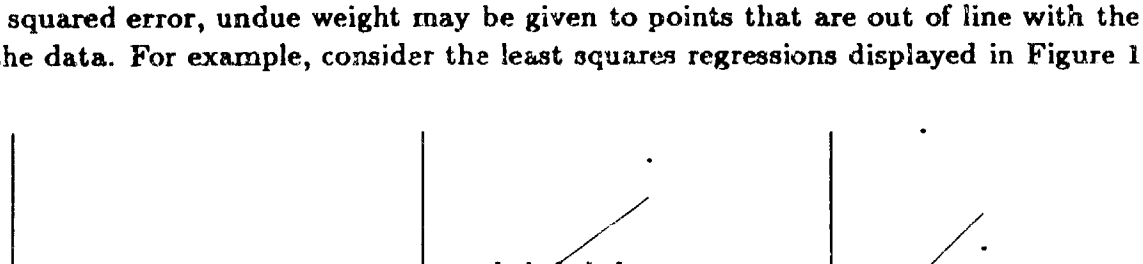

Figure 1: Least squares linear regression when some of the data is corrupted.

There are a variety of ways of dealing with this type of problem. One method which retains all the data is based upon M-estimates.

Regression M-estimate: In least squares regression the objective is to minimize the sum of the squared errors. This gives great importance and control to those data points that normally would have a large error. If the sum of the squares is replaced by a less rapidly increasing function of the residuals an M-estimate is produced. The problem then becomes

$$\min_{\beta} \sum_{i=1}^{n} \rho(z_i \quad \mathrm{x}_i, \beta)$$

for some cost function $\rho(\cdot)$. The minimum can be characterized by the follcwing equations:

$$\begin{aligned} \sum \psi(\frac{z_i - \mathbf{x}_i\hat{\beta}}{\sigma}) X_{ik} &= 0, \quad k = 0,1,2 \\ \sum \chi(\frac{z_i - \mathbf{x}_i\hat{\beta}}{\sigma}) &= 0, \end{aligned}$$

where

$$\begin{aligned} \psi(x) &= \rho'(x), \\ \chi(x) &= x\psi(x) - \rho(x) \end{aligned}$$

For the test case used here, the cost function switches from square to linear once the residual becomes too large. In particular

$$\psi(x) = \max(-c, \min(c, x)) \quad \text{with } c = 1.5.$$

The constant c was chosen based upon the experience of experts in linear surface estimation. Additional testing of this technique on LADAR data is needed for an optimal setting. Since this is a nonlinear function, the regression must be solved by iteration. The algorithm used is detailed in Huber [5]. For more specifics on the asymptotic behavior of M-estimates in regression see Huber [6] as well.

Another technique for dealing with data points that are suspected of being corrupted is to remove them entirely from the least squares computation. First though, these data points must be identified as outliers.

Outlier identification: The identification of outliers in a sample usually proceeds via a variety of tests with a good deal of human subjectivity involved. Most methods start with a least squares regression using all the points. Then the residuals from that regression and knowledge of the design matrix X are used to find outliers.

The residuals from least squares,

$$\hat{\mathbf{e}} = \mathbf{z} - X\hat{\beta} = (I - X(X^TX)^{-1}X^T)\mathbf{z} = G\mathbf{z},$$

have a variance that changes according to the location of $\mathbf{x}_i$. For example, the addition of a constant offset to all the data samples will affect the residuals $\hat{\mathbf{e}}$. For this reason it is not wise to compare the residuals directly with each other. Instead there is a "normalized" version of the residual,

$$r_i = \frac{\hat{e}_i}{\sigma\sqrt{g_{ii}}}$$

which has a variance of 1 so that the values are comparable. If the variance of the noise, σ^2, is not known, then it can be estimated from the data as

$$\hat{\sigma}^2 = \frac{\hat{\mathbf{e}}^T\hat{\mathbf{e}}}{n-3}$$

A large normalized residual is not the only indicator of a bad data point. In the first plot in Figure 1 the residual at the last point is not very large but it is still an outlier. This point has undue influence on the results of the regression and is called a leverage point. The values on the diagonal of the matrix G, g_{ii}, are related to the leverage at point i. As g_{ii} approaches 0, the leverage increases.

There are several ways of combining the values of r_i and g_{ii} into an influence function that measures the effect of point i on the total regression as compared to the other points. Two of them are:

(1) Normalized residual $\quad H_1(r_i, g_{ii}) = r_i^2$, and

(2) Cook's function $\quad H_2(r_i, g_{ii}) = \frac{r_i^2}{3}(\frac{1 - g_{ii}}{g_{ii}})$

Other examples of influence functions are given in [3].

Those data points which have abnormally large values of the influence function are removed from the data set and least squares regression is repeated without them. This process can be repeated as needed. Several questions arise when using this strategy:

- What is meant by "abnormally large"?
- How many points can be removed at once?
- When should the process be terminated?

The variance of the normalized residual is one and the mean is zero, so if the errors are all gaussian we will have $|r_i| < 2.57$ with probability .99. Therefore a value of H_1 greater than 6.6 is highly suspect. Cook's function does not have a pre-determined variance which is constant over all the data. It measures the effective change in $\hat{\beta}$ that would occur if the i-th data point were removed. The removal of most points will have little effect on the regression estimate. Those points whose removal causes a change that is significantly larger than average are potential outliers. These points are identified by estimating the variance, s^2, of $r_i\sqrt{(1 - g_{ii})/g_{ii}}$ over the sample data. When $H_2 > 6.6s^2$ there is probably an outlier present.

For the highest accuracy, it would be best to remove only one point during each pass. This allows the residuals to readjust and avoids forcing the estimate to remain the same because all non-supporting data has been removed immediately. This is a very time consuming process, however, so there is a tradeoff between time and the potential for ruining the computation. The potential for eliminating a large set of correct points is avoided if the number of points that can be removed per pass is fixed to a certain percentage of the points currently present. In these tests at most 10% of the points are targeted for removal in the each pass.

The outlier removal can continue until no more points meet the outlier criterion. This is a dangerous practice, however, since the number of points can be become so small that the results are meaningless. It is important therefore to limit the total number of points that can be eliminated, as well as the number of outliers removed at each stage. For these

tests, the limit on total number of points removed is based upon the size of the perimeter — since it will often be the perimeter points which are suspect.

Tests

In order to test the planar estimation routines described above, several sets of synthetic data were generated. Parameters were varied to produce the data sets. These parameters included:

- *Image Size:* Either 5x5, 10x10, or 15x15.
- *Expected Noise:* $\sigma = 1$, or 5.
- *Extra Corruption:* 0% or 10% of the points had added noise with standard deviation $s = 20$. The corruption was either Gaussian or a one-sided Gaussian.
- *Angle of Plane:* Either perpendicular to the line of sight 0° or at 45°.

The specifics of the different data sets are contained in Table 1.

Each of the fitting techniques described in the previous sections was run on all the synthetic data. Table 2 shows the labels that are used to denote the different algorithms. To score each run, the interior angle between the computed normal and the true normal was calculated. The closer this angle, θ, is to zero the better the result. The scores from each of the runs are shown in Tables 3 and 4. The different algorithms were then compared by using the Wilcoxon signed-ranks pairwise hypothesis test (see PM II monthly report August 1986).

It should be noted that other tests that were considered, but were not performed due to lack of time. For instance, a test of data which had corrupted pixels along one boundary, simulating the effect of slight missegmentations, could have been performed.

Conclusions

First it should be noted that when the noise was increased to $\sigma = 5$, as in data set DS4, the planar fits for a 5x5 region were so distorted that they were useless. Because of this, future tests with $\sigma = 5$ were not performed on the 5x5 images and the results from the 5x5 images in DS4 were ignored. Also a check of the estimated means and variances of θ between data sets DS1, DS2, and DS3, and their counterparts in DS7, DS8, and DS9 shows that the viewing angle of the plane does not significantly alter the behavior of the algorithms. The slight improvement in the angular results when the viewing angle is 45° is due to slope to angle transformation rather than any actual performance improvement. The number of pixels which can be seen on the plane is a much more important factor in determining algorithm performance. Generally of course, for a fixed size surface, the number of viewed pixels will be proportional to the viewing angle.

Table 5 presents the results from the pairwise hypothesis tests, and the conclusions are summarized here. Each pairwise test that is performed has a potential "winner" and

"loser". When the significance level becomes small enough, there is evidence to support this winner-loser division. If the level is too large, neither algorithm can be called a "winner". In fact a result as high as 0.3 is meaningless, and comparisons that produced such results are not presented in the table. On the other hand, the appearance of an algorithm pair in the table does not guarantee that one is better than the other. Confidence in this type of conclusion must be based upon the significance level. For instance, in data set DS1 we see that the M-estimate (M) performed better than Normalized residual outlier removal (N) and Least Squares (LS) with a fair amount of certainty, and better than Cook's function outlier removal (C) with less certainty. But, no one of N, LS, or C can be said to have performed better than any of the others.

A brief scan of the results of the Wilcoxon hypothesis tests on the separate data sets, shows that the M-estimate is often the top performer, while least squares is often the worst. As the noise increases the least squares estimate appears to perform relatively better than at low noise levels. In fact when only the expected noise is present least squares is the best performer, but it is still at a disadvantage when there is extra corruption in the image beyond the expected noise. Since least squares is the maximum likelihood estimate when the noise distribution is truly Gaussian it must be concluded that quantization has significantly distorted the distribution. This is hardly surprising in the case where $\sigma = 1$. The performance of the M-estimate depends upon the value of the threshold, c, which remained fixed in these tests. Ideally this threshold would vary based upon the characteristics of the input data. (There are allusions to this in the literature, but we have not yet seen any specifics.) Therefore, the M-estimate may perform even better than is demonstrated in these experiments.

When all the data sets are merged together a very definite ordering of algorithm performance appears.

1. M-estimate.
2. Normalized residual outlier removal.
3. Cook's function outlier removal.
4. Least squares.

When using parametric surface estimation alone, M-estimates is the most robust of the four techniques tested. M-estimates is iterative and can be time consuming, so it might be possible to get similar results by (i) using Gaussian smoothing to remove part of the quantization error, (ii) applying a replacement type filter to remove extreme points, and (iii) proceeding with least squares.

Table 1: Data sets used for planar estimation tests.
(See text for options.)

Data Set	Angle	Expected Noise	Gaussian Corruption	No. of Images/Size		
				5x5	10x10	15x15
DS1	0°	$\sigma = 1$		5	5	5
DS2	0°	$\sigma = 1$	10%	5	5	5
DS3	0°	$\sigma = 1$	10% one-sided	5	5	5
DS4	0°	$\sigma = 5$		5	5	5
DS5	0°	$\sigma = 5$	10%		5	5
DS6	0°	$\sigma = 5$	10% one-sided		5	5
DS7	45°	$\sigma = 1$		5	5	5
DS7	45°	$\sigma = 1$	10%	5	5	5
DS7	45°	$\sigma = 1$	10% one-sided	5	5	5

Table 2: Labels of the plane fitting algorithms.

LS:	Standard least squares.
M:	Robust M-estimate regression.
N:	Outlier removal based on normalized residuals.
C:	Outlier removal based on Cook's function.

Table 3: Angle, θ (in degrees) between result normal and actual normal. The mean, m, and estimated variance, sd, of the angle are given for each size grouping.

		LS	M	N	C		LS	M	N	C		LS	M	N	C
5		5.71	4.50	5.55	5.71		63.43	24.28	19.21	21.59		28.82	0.38	4.04	4.80
x		11.53	13.12	13.24	13.24		56.24	26.87	18.27	17.67		48.02	17.84	28.19	18.17
5		4.57	1.57	4.57	4.57		34.59	0.19	12.90	12.51		22.99	23.40	25.24	25.24
		18.53	15.98	18.57	18.57		41.61	13.34	9.50	9.50		62.40	19.03	16.98	19.98
		12.75	9.10	12.75	12.75		31.31	14.14	14.76	14.76		49.75	16.51	24.01	31.13
m		10.02	8.85	10.94	10.97		45.48	15.76	14.93	16.81		42.40	15.39	19.69	19.86
sd		4.7	5.94	5.84	5.80		13.88	10.57	3.97	4.70		16.18	8.80	9.67	9.82
10		4.60	3.31	3.81	2.06		16.30	4.03	3.25	4.70		4.03	0.85	1.86	1.86
x		2.24	2.23	1.18	1.45		19.73	1.80	1.23	1.66		11.61	4.69	4.57	5.52
10	DS1	1.94	0.04	3.03	2.49	DS2	29.56	5.83	1.24	2.47	DS3	8.00	5.62	6.08	4.25
		2.67	2.95	2.67	2.57		37.45	0.95	21.50	35.23		22.42	1.15	1.63	1.63
		2.33	2.72	1.04	1.78		19.46	1.37	3.30	3.78		12.26	3.59	1.49	1.65
m		2.76	2.25	2.35	2.07		24.5	2.80	6.10	9.57		11.66	3.18	3.13	2.98
sd		1.06	1.30	1.20	0.47		8.78	2.07	8.67	14.39		6.85	2.12	2.08	1.80
15		0.35	0.60	0.24	0.24		0.32	1.20	1.20	1.14		5.96	0.79	0.98	1.01
x		1.70	0.78	0.87	0.65		1.31	0.52	0.71	1.17		11.03	1.80	7.53	7.99
15		1.40	1.04	0.87	1.73		4.59	0.56	1.07	0.85		7.71	2.40	1.61	3.32
		0.68	0.30	0.35	0.19		6.44	2.80	4.61	5.77		4.64	3.62	2.12	3.73
		0.30	0.27	0.62	0.72		8.90	2.70	1.05	1.57		7.94	0.34	2.65	4.82
m		0.89	0.60	0.59	0.71		4.31	1.56	1.73	2.10		7.46	1.79	2.98	4.17
sd		0.63	0.33	0.29	0.62		3.56	1.12	1.62	2.07		2.41	1.31	2.62	2.55
		LS	M	N	C		LS	M	N	C		LS	M	N	C
5		63.07	62.66	62.27	63.07										
x		34.59	44.41	34.59	34.59										
5		52.59	56.53	52.59	45.52										
		32.53	36.83	43.76	43.92										
		53.17	48.42	53.17	59.69										
m		47.19	49.77	49.28	49.36										
sd		13.14	10.12	10.50	11.80										
10		15.55	20.77	21.41	29.03		15.10	8.74	3.86	4.70		11.01	15.19	27.66	24.80
x		6.54	13.23	3.52	12.67		18.22	22.70	26.79	23.37		15.92	2.68	9.41	10.58
10	DS4	21.02	23.74	34.89	33.61	DS5	11.43	4.50	13.96	10.93	DS6	21.70	23.20	36.24	33.06
		10.07	14.67	12.54	12.01		15.84	9.08	6.61	2.49		25.21	18.82	15.25	7.93
		15.80	23.68	34.25	24.94		11.86	8.16	19.35	26.41		9.50	11.63	13.35	13.30
m		13.80	19.22	21.32	22.45		14.49	10.64	14.11	13.58		16.67	14.30	20.38	17.93
sd		5.61	4.98	13.65	9.73		2.85	6.89	9.35	10.83		6.75	7.79	11.19	10.63
15		5.98	4.32	5.50	9.34		13.27	7.05	8.67	12.83		9.47	6.06	7.19	7.05
x		2.72	4.87	4.98	5.15		7.33	8.15	2.73	4.19		3.52	5.26	6.77	7.61
15		5.52	5.24	6.54	5.31		4.23	3.55	13.58	9.33		6.66	3.76	7.14	10.45
		6.57	4.60	5.36	4.75		19.17	17.22	17.16	15.85		3.07	2.94	7.13	7.26
		5.15	3.34	1.67	3.96		2.60	9.15	8.51	10.34		14.82	12.26	8.63	11.79
m		5.19	4.47	4.81	5.70		9.32	9.02	10.13	10.51		7.51	6.06	7.37	8.83
sd		1.48	0.72	1.85	2.10		6.85	5.05	5.50	4.34		4.84	3.68	0.72	2.15

Table 4: Angle, θ, (in degrees) between result normal and actual normal (continued).

		LS	M	N	C		LS	M	N	C		LS	M	N	C
5		5.16	7.37	5.16	5.16		27.90	12.82	15.00	12.96		57.16	3.62	4.20	4.20
x		8.46	10.87	8.46	10.43		12.45	3.28	14.53	2.95		72.18	4.91	7.07	7.07
5		3.70	4.03	3.70	3.70		2.61	5.29	4.33	8.06		10.39	4.44	7.85	10.39
		12.11	15.21	16.18	16.18		68.43	9.47	14.88	11.83		3.12	0.05	3.12	3.12
		1.07	3.09	1.07	1.53		6.07	11.52	10.96	13.81		12.16	12.64	11.63	11.63
m		6.10	8.11	6.91	7.4		23.49	8.48	11.94	9.93		31.00	5.13	6.77	7.28
sd		4.29	5.01	5.83	5.9		26.93	4.07	4.57	4.47		31.37	4.61	3.35	3.72
10		0.67	2.02	2.07	2.28		7.88	0.90	1.20	1.57		3.81	0.31	4.47	0.73
x		0.91	0.69	0.91	1.58		3.56	0.43	1.20	1.26		4.19	1.07	0.88	0.85
10	DS7	1 20	1.59	0.94	1.13	DS8	8.31	0.19	1.33	1.33	DS9	14.53	0.18	1.12	1.50
		1.79	0.15	1.36	0.98		8.45	0.67	0.28	2.02		6.20	2.23	1.46	1.97
		2.09	2.53	1.35	2.66		2 88	2.02	3.32	2.34		2.69	1.29	1.67	1.45
m		1.33	1.40	1.33	1.73		6.22	0.84	1.47	1.70		6.28	1.02	1.92	1.30
sd		0.35	0.97	0.47	0.73		2.75	0.71	1.12	0.46		4.78	0.83	1.46	0.51
15		0.54	0.28	0.17	0.27		6.64	0.78	1.34	1.05		3.47	0.06	0.42	0.69
x		0.33	0.54	0.60	0.91		3.85	0.20	1.10	1.60		2.10	0.45	0.13	0.09
15		0.44	0.24	0.49	1.14		3.19	0.42	0.62	0.32		0.90	1.16	0.54	0.85
		0.32	0.16	0.35	0.62		1.28	0.84	1.13	1.48		7.13	0.79	0.40	0.47
		1.15	1.43	1.17	1.23		2.05	0.33	0.24	0.23		3.55	0.26	0.36	0.45
m		0.56	0.53	0.56	0.83		3.40	0.51	0.89	0.94		3.43	0.54	0.37	0.51
sd		0.34	0.52	0.38	0.39		2.07	0.28	0.45	0.64		2.34	0.44	0.15	0.29

Table 5: Results of the Wilcoxon signed-ranks hypothesis tests. Each triple consists of: the "winner", the "loser", and the significance level. Smaller significance levels indicate higher confidence in the "win-lose" decision. Triples with levels ≥ 0.3 are not shown.

DS1			DS2			DS3			DS4			DS5		
M	C	0.140	N	C	0.010	M	N	0.056	LS	M	0.037	M	N	0.222
M	N	0.096	N	LS	0.000	M	C	0.012	LS	N	0.142	M	C	0.254
M	LS	0.070	C	LS	0.001	M	LS	0.000	LS	C	0.014	M	LS	0.101
			M	LS	0.000	N	C	0.056	M	N	0.254			
						N	LS	0.001	M	C	0.046			
						C	LS	0.001						
DS6			DS7			DS8			DS9			All together		
M	LS	0.142	LS	M	0.018	M	N	0.015	M	N	0.116	M	N	0.002
M	N	0.046	LS	C	0.018	M	C	0.002	M	C	0.086	M	C	0.000
M	C	0.046	N	M	0 128	M	LS	0.003	M	LS	0.001	M	LS	0.000
LS	C	0.222	N	C	0.006	N	LS	0.010	N	C	0.197	N	C	0.004
						C	LS	0.012	N	LS	0.001	N	LS	0.000
									C	LS	0.001	C	LS	0.000

2.4 Ground Plane Removal

The thesholds used for the region growing segmentation are computed directly from the image, but experience has shown that they are set inappropriately when a large amount of background is included in the computation. This is to be expected since the background is quite often not as smooth and regular as the target itself. Therefore, there is a need to eliminate most of the background in an image before computing the surface fitting thresholds.

The goal of this effort was *not* to totally segment the target object from the background, but to identify areas that can be reasonably expected to belong to the background. Two major techniques were investigated in this regard.

- Create closed boundaries from edge pixels.
- Model the background as a plane and strive to find this plane.

The first technique used edge information along with edge linking to try to produce closed boundaries. This was found to be unreliable since linking was rarely complete and the lower edge between target and ground plane was often not detected. The linking techniques based upon edge magnitude and direction may prove useful though when trying to identify crease edges. Figure 2 shows an example of the edge detection on synthetic noise free data. The failure to detect the bottom edge of the target occurs even in this noise free environment.

The second technique, based on a background model rather than edges, was more effective. One of the largest major components of the background is the ground plane. Therefore efforts were directed at finding and identifying a planar region that occupies a major portion of the boundary of the image, and extrapolating this information to the interior of the image.

Two different approaches to finding and removing the ground plane were investigated. The first involved simple accumulation of evidence. It was assumed that the majority of the boundary of the image is part of the background. Therefore a planar fit was computed for a boundary point, and then any point within the image that lay close to this plane accumulated evidence that it was also a part of the ground plane. Once this was done for all the boundary points in turn, those points with the highest accumulation of evidence would be background points. Unfortunately, this technique was so time consuming relative to the surface extraction itself that it was judged infeasible.

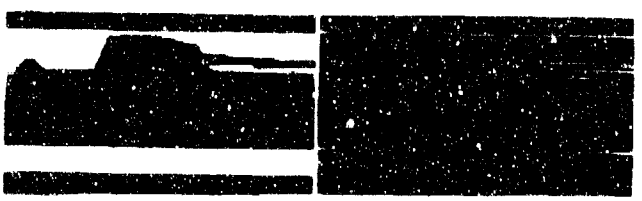

Figure 2: Edge detection on a synthetic range image of a T-55 tank.

The second approach to removing the ground plane is based on similar assumptions, but uses clustering techniques instead of accumulation. Briefly, the planar normal and intercept are approximated at each point in the image and the normals are clustered to find parallel planar regions. The cluster most likely to contain the background plane is then separated based upon intercept value so that object regions parallel to the backgound plane are not removed. This approach is much faster than the previous one, and is described in more detail in what follows.

The normal at each point is represented by the x and y partial derivatives and the intercept is computed (mod 256) using the center of the image as the origin. The normals are then clustered in two-dimensional space using a convergent k-means technique. Since we want to separate out one particular area in 2-D space, a 5-means clustering was used. This allows for clusters on each side of the principal cluster. Brief tests with other numbers of means have all produced inferior results. The clustering routine currently available employs the Euclidean norm, but we would expect improved results if the Mahalanobis norm were substituted.

In order to determine which of the clusters contains the ground plane the assumption of backgound occupying the majority of the boundary is again used. Other types of information such as the depression angle or terrain maps could be used to enhance this determination. Once this clurter is identified it is further divided based upon the intercept values. The intercepts are histogrammed to find the major background intercept region, so that the other parallel areas can be removed from consideration. This is done by finding the maximum histogram value and moving outward until the value drops below a given percentatge of the maximum. Points with the appropriate normal and intercept value which are connected to the boundary are identified as potential background points. Finally those areas not currently identified as backgound are classified as either an area of interest based upon some external criterion or added to the background region.

Figures 3 and 4 show various stages of the ground plane removal performed on synthetic imagery. The original image in Figure 3 is a T-55 tank range image generated from ERIM wire frame model with ground plane added at a depression angle of 10°. Gaussian noise was added to the image to create SNR = 4. The sequence shows the 5 different normal clusters, the chosen background cluster with points removed based upon intercept value, the connected portion of the background, and finally the foreground area which will be handed off for segmentation.

The original image in Figure 4 is an M-109 viewed at a depression angle of 10° and the corresponding ground plane. No noise was added to this image. In this case, the sequence shows the background cluster both before and after separation based upon the intercept, and the mask to be used for segmentation. The only substantial differences between the final mask and the original object are the artifacts due to the range ambiguities. The mask is slightly larger than the real object because of averaging effects.

To give an idea of the power of the surface segmentation, Figure 5 shows the segmentation of the original noise free T-55 image and the M-109 image. The sawtooth edge occuring in the M-109 segmentation is due in part to quantization effects. The gun barrel of the T-55 is included as part of one of the turret planes since the narrow barrel produces

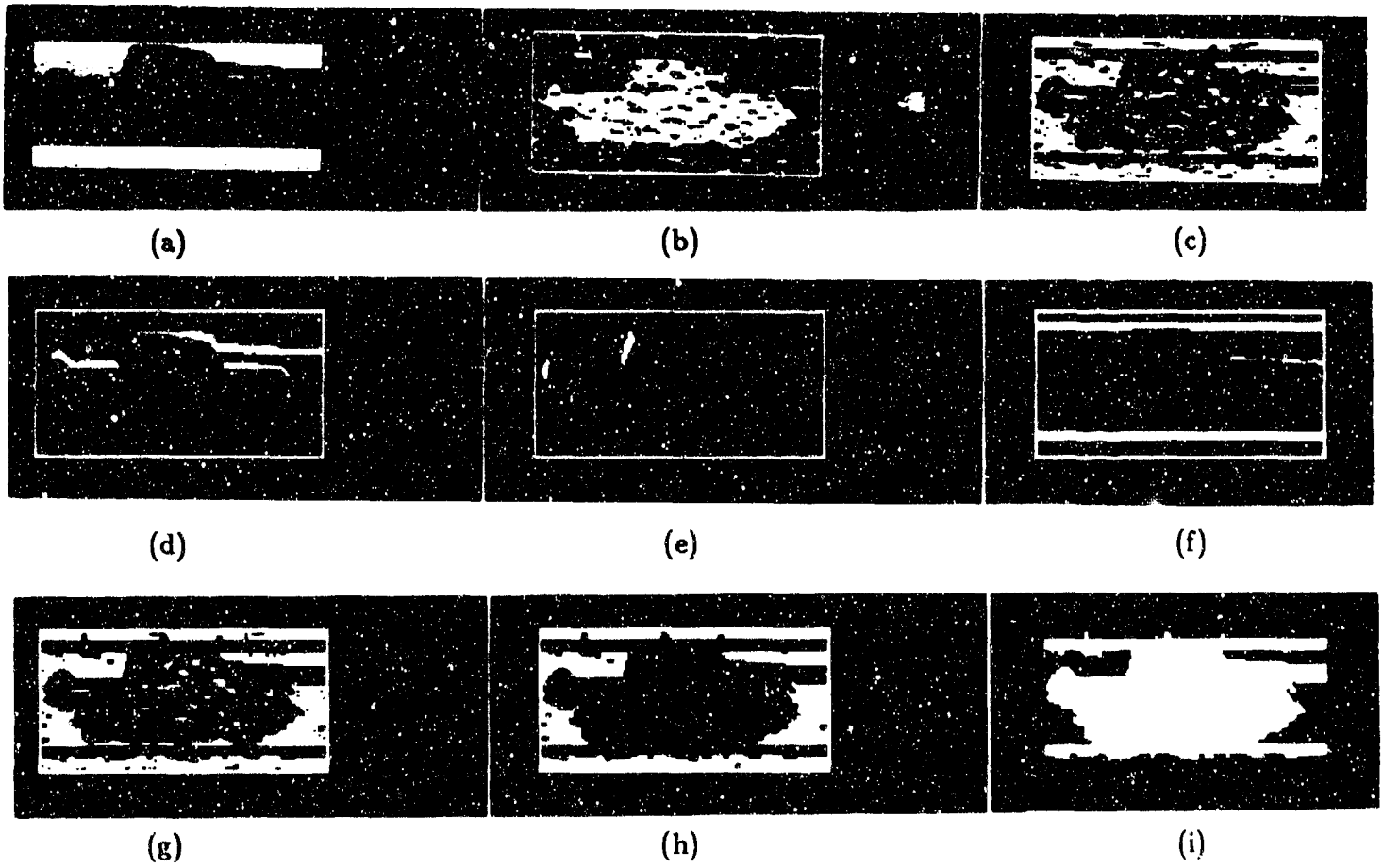

Figure 3: Ground Plane Removal: (a) original image: T-55 at SNR = 4, gun barrel clipped to reduce image size, (b-f) points included in each of the 5 normal clusters, (g) points in ground plane cluster minus those with the wrong intercept values, (h) connected component of ground plane, (i) final mask identifying foreground.

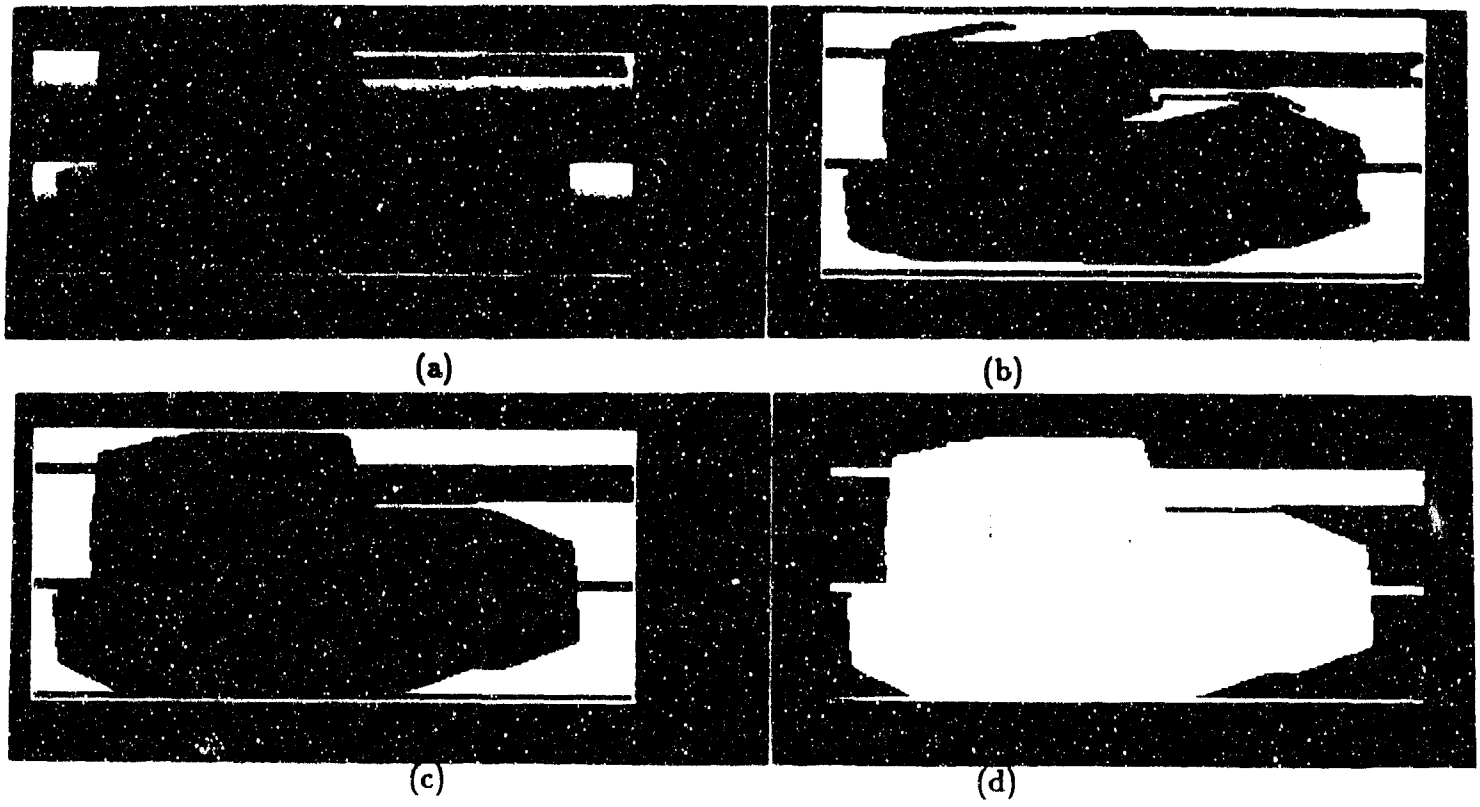

Figure 4: Ground Plane Removal: (a) original image: M-109 without noise, (b) ground plane cluster, (c) after removal of points based upon intercept, (d) final mask of foreground.

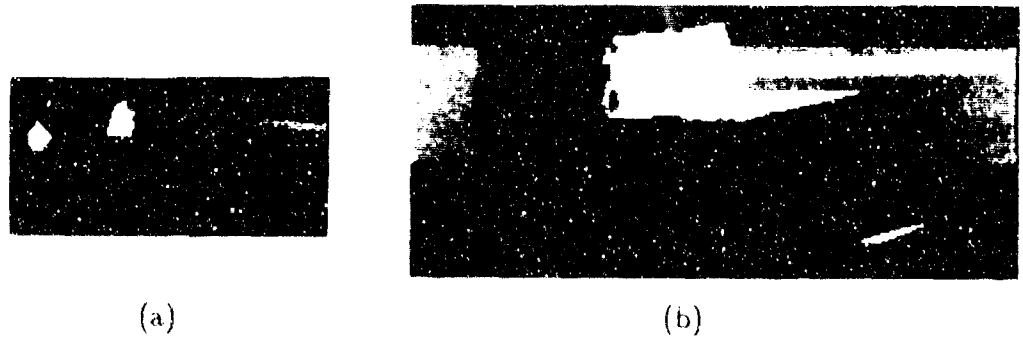

Figure 5: Surface Segmentation Examples without noise: (a) T-55, (b) M-109

degenerate planar fits by itself. This might be corrected by also allowing quadratic fits.

2.5 Smoothing Prior to Computing Derivatives

First and second order partial derivatives are often used in range image analysis, but measuring them reliably is difficult. Edge detection may be viewed as the characterization of intensity changes within an image. Thus, effective edge detection necessitates the reliable estimation of the differential behavior of the image function. Specifically, derivatives of various types, order and scale have to be accurately calculated for edge detection to function correctly. Surface normals and Gaussian and mean curvature are also functions of partial derivatives and thus have the same reliability problems.

Numerical differentiation of a sensed image function is an ill-posed problem because its solution does not depend continuously on the data. This may be directly attributed to the random perturbations introduced by the previous sensing and sampling processes. Thus, before edge detection can be reliably performed, the differentiation problem must become well-posed and stable.

As argued in [11] ill-posed problems can be be solved using the regularization methodology originally suggested in [9,15]. Briefly, the general approach is to restrict the space of possible solutions to subspace by imposing supplementary constraints. Of the many possible regularization techniques, one that is natural for edge detection on range images is that of Reinsch [12] considered for numerical differentiation. In the context of edge detection, this approach suggests optimal filters and their approximations that can be applied to the noisy data prior to computing derivatives [11].

In the literature, various criteria have been proposed for directing the determination of "optimal" regularization filters. Basically, these criterion center upon a tradeoff between the degree of regularization of the filtered data versus the closeness of filtered data to the original data. Once the criterion has been selected, a regularization filter must be determined for each specific type of derivative.

For jump edges, a Gaussian operator is a good approximation [11] to the optimal filter derived using a calculus of variations technique. Although the regularization approach suggests the near optimality of Gaussian filters, the variance of the Gaussian function corresponding to the scale of the filter must be specified. At least two or three methods for determining the scale have been suggested in the computer vision literature [11,18] but we are not aware of any that have been applied to infrared or range images.

In addition to finding the optimal scale for a Gaussian filter there is the option of applying the filter at various scales. In an image, changes of intensity take place at many spatial scales depending on their physical origin. Consequently, a multiscale analysis, tracing the behavior of some feature of the signal across scales, can reveal useful information about the nature of the underlying physical process. For example [19], spatial coincidence at all scales of zero crossings in the Laplacian of Gaussian filtered images may confirm a physical "edge" distinct from surface markings. It should be emphasized that it is not only necessary to detect and describe changes in a range image at different scales, but in addition, much useful information can be obtained by combining descriptions across scales.

Currently a Gaussian filter is being applied to the range data prior to computing the derivatives needed for Gaussian and mean curvature. The results of this filtering are particularly obvious on synthetic data. Quantization effects can easily be seen on the *HK*-Sign map for a plane which is at an angle to the line of sight. Instead of labelling the entire plane as a flat, periodic stripes occur across the plane of other surface types. Regularization filtering helps to minimize this quantization effect as well as suppressing the noise.

2.6 Crease Edge Detection

During surface segmentation it is possible that a single surface becomes subdivided into several separate regions. If these separate regions are used when forming the object graph then any matching will be unlikely to succeed. Therefore a method must be found for deciding when two regions belong to the same surface as opposed to two separate surfaces.

We are proposing to make decisions on the distinctness of two regions based upon the presence or absence of crease edges. If a crease edge is present then the two regions remain separate. Otherwise the two adjacent regions are merged and considered as one surface.

Jump edges tend to be easily identified by the surface growing algorithm, but crease edges are harder to find. Mitiche and Aggarwal [8] have proposed a method to crease edge detection which has the advantage of being based upon a known noise model. This technique has been implemented and is in the process of being tested and integrated with the surface segmentation software.

This approach is based on fitting planes in various directions in a local neighborhood and computing the angles between the planes. Let $g(x_0, y_0)$ be the depth map value at (x_0, y_0) and N be an appropriate sized neighborhood around (x_0, y_0). Let N be divided into two contiguous planar surfaces, c_1 and c_2, n_1, and n_2 being the surface normals of the planes best fit (say in least square sense) to the points in c_1 and c_2 respectively. The collection of points in N together with these best fit planes will be referred to as a partition of N and is denoted as R. Suppose n is the number of directions in which crease edges have to be determined. In our implementation n is 4, corresponding to horizontal, vertical, left diagonal, and right diagonal directions. For each direction, planes are fit according to some selected criterion to the corresponding two regions in N thus determining the partitions R_i, $i = 1, \ldots, n$. The angles between the planes associated with these partitions are denoted by Θ_i, $i = 1, \ldots, n$. The crease edge detection algorithm then proceeds as follows:

(i) All points with $\Theta_i < t$, $i = 1, \ldots, n$ are discarded for some t. This step is intended to eliminate deep "interior" points that are surface points far from edges or jump boundaries.

(ii) The likelihood of each of the partitions at the remaining points (denoted by set Ω) is computed as follows:

$$p(g|R_j) \propto \prod_{(x_0,y_0)\in\Omega} p(g(x_0, y_0)|R_j), \tag{1}$$

where $p(g(x_0, y_0)|R_j)$ is a function of the physical properties of the target (surface orientation with respect to the signal beam direction, reflectance and actual distance from the sensor) and sensor parameters (fixed for a given scene).

Although it would be ideal to derive $p(g(x_0, y_0)|R_j)$ from the physics of the range image function, it is a very complicated problem that is not currently addressed. Presently we are modelling the function $p(g(x_0, y_0))$ as normally distributed with mean μ = distance D to the estimated plane at point (x_0, y_0), and constant variance. If reliable reflectance data is available then the variance model can be augmented to $\sigma(x_0, y_0) = \sigma_{(x_0,y_0)}(D, \rho, \phi)$, where ρ is the surface reflectance and ϕ is the orientation of the appropriate plane R_j with respect to the signal beam direction. When σ is constant the likelihood function reduces to the root mean square error of the two planes combined.

Once the likelihood values are computed, the partition with maximum likelihood (or minimum RMSE) is selected. Then all points for which $\Theta < t$, where Θ is the angle associated with the best partition, are identified as crease edge points.

A more implementation-oriented description of the software we are currently using for crease edge detection is given here.

1. At each point an edge with a particular orientation (horizontal, vertical, either diagonal) is hypothesized. Pixels on either side of this edge are fitted to two separate planes.
2. The root mean square error of the computed planar fit is calculated.
3. When the fits and RMSEs have been computed for each of the edge directions, then the edge direction with minimum RMSE is retained as a potential edge.
4. Potential edges are then accepted or rejected based upon an RMSE threshold and local minimum filtering.
5. The angle between the two surfaces is computed for each of the accepted edges. If this angle is less than a certain angular threshold then the edge is considered nonexistent.

This technique will find some jump edges as well as crease edges so an additional test has been added to remove jump edges based upon a jump distance computed between the two planes at the point in question. This jump distance is best described with reference to Figure 6. The point where the jump distance is to be computed is labelled x. The planes P_1 and P_2 on either side of the proposed edge cross position x at points z_1 and z_2 respectively. The jump distance is the average of the shortest distance from point z_1 to plane P_2 and likewise the shortest distance from z_2 to P_1. When two planes are parallel this distance is identical to the shortest distance between the planes. This measure is much more dependent on relative planar orientation than on absolute orientation which makes it more useful than other jump distance measures.

An example of the behavior of crease edge detection prior to thresholding is given in Figure 7. The first frame is the base image that is to be analyzed. The surface segmentation of this image was included in the September monthly report. Figure 7-b shows the

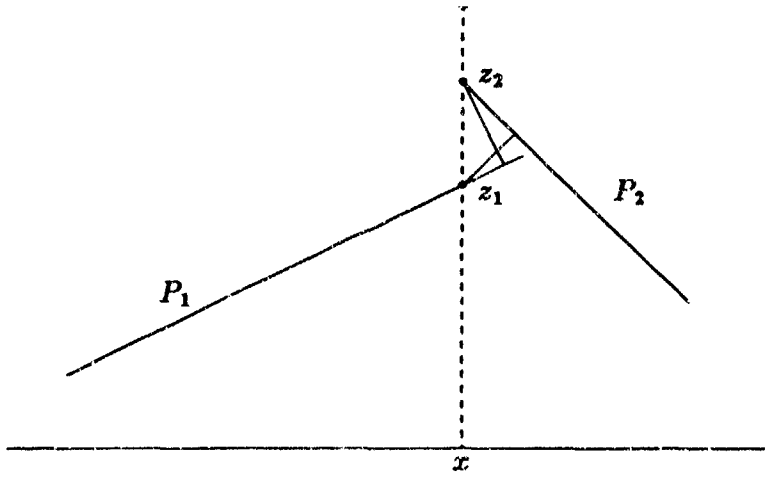

Figure 6: Computing the Jump Distance.

minimum RMSE values computed at each pixel. Bright areas indicate a large amount of error and are not good candidates for crease edge pixels. Figure 7-c shows the angles between the planes of best fit. Angles which are too small indicate no crease edge. Finally the last frame shows the jump distance corresponding to this fit. Large jump values also eliminate the possibility of a crease edge. Inspection of these images indicates that the most likely place for crease edges is at the lower edge of the target, between the body of the tank and the turret, and within the turret itself.

Since we do not have any accurate noise models for the laser range data the noise model assumed in this implementation is additive Gaussian. This also allows for easy simplified testing on synthetic data. When the correct noise model for the data is known, it can be used to tailor the likelihood function.

When crease edge detection is being used in conjunction with the surface fitting it is not necessary to test for crease edges over the entire image. The crease edge detection can be restricted to those areas in the immediate vicinity of surface intersections that result from the surface growing. The surface growing algorithms also use thresholds that are related to the likelihood of a planar fit. These thresholds are computed initially from those portions of the image that are expected to contain the target. The threshold used during the surface growing is expected to be highly correlated with, if not identical to, the threshold of the likelihood function mentioned above.

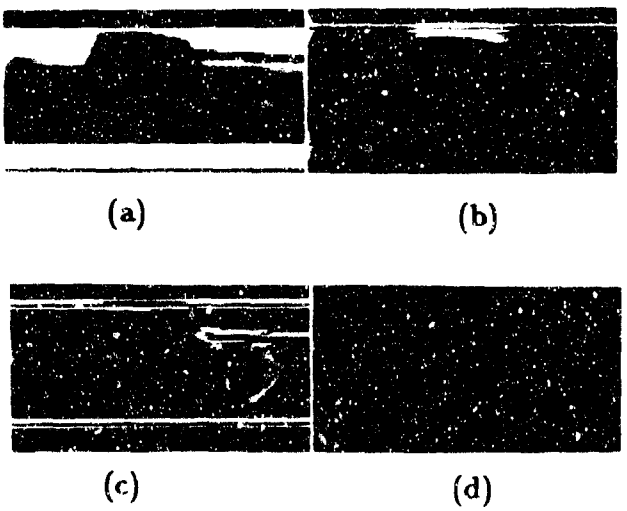

Figure 7: Crease edge example: (a) original image, (b) error for the plane fit associated with best edge orientation, (c) angle between planes, (d) jump distance between planes.

3.0 SUMMARY AND FUTURE WORK

The progress made in the 15 May 1987 to 15 November 1987 reporting period concerned preliminary implementation of surface segmentation/description techniques. A modified version of Besl and Jain's region growing segmentor is in place and shows promising results. This is a major step towards developing an effective LADAR target recognition methodology.

In the previous reporting period (November 1986 – May 1987) a preliminary simplified version of the matching algorithm was implemented. Future work will be aimed at creating the links between these two algorithms. The major tasks involved in this effort include:

- Integration of the jump and crease edge detection into the surface segmentation, so that a final segmentation image is produced composed of known planes.
- Smoothing and multiscale analysis will be investigated further for handling the question of variable resolution.
- Description stage: Features of each of the surfaces and their relationships must be computed. Any surfaces thought to belong to the background must be removed at this stage. The feasibility and utility of curve description will be investigated as well.
- Matching: The matching routines must be upgraded to handle a larger feature set, occlusion, and varying ranges. Orientation approximation routines will be added to eliminate the need for multiple target views in the database. The utility of different features as an aid to matching will be studied.
- Database: A search must be made for imagery that more accurately approximates the expected working data. A major problem in testing has been the scarcity of data representative of the specifications that were originally assumed. It would be useful to be able to run tests on data which is free of range ambiguities and is not range-gated.
- Modelling: Graph models for several target types will be developed based upon the ERIM wireframe models.

REFERENCES

[1] P.J. Besl and R.C. Jain. Segmentation through symbolic surface descriptions. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 77–85, IEEE, Miami Beach, Florida, June 22–26, 1986.

[2] P.J. Besl and R.C. Jain. Three-dimensional object recognition. *ACM Computing Surveys*, 17(1):75–145, March 1985.

[3] A. Dempster and M. Gasko-Green. New tools for residual analysis. *Ann. Statist.*, 9(5):945–959, 1981.

[4] W.G. Hanley and M.A. Lepley. Performance Modelling of Autonomous Electro-Optical Sensors, Technical Progress Report on Contract #DAAB10-86-C-0534, February 1987.

[5] P.J. Huber. Robust regression: asymptotics, conjectures and Monte Carlo. *Ann. Statist.*, 1(5):799–821, 1973.

[6] P. Huber. *Robust Statistics*. John Wiley and Sons, New York, 1981.

[7] M. Lepley, W.G. Hanley, J.E. O'Connor, and R. Chellappa. Performance Modelling of Autonomous Electro-Optical Sensors, Second Semi-Annual Report on Contract #DAAB10-86-C-0534, May 1987.

[8] A. Mitiche and J.K. Aggarwal, Detection of Edges Using Range Information, *IEEE Transactions of Pattern Analysis and Machine Intelligence*, PAMI-5, pp. 174–178, March 1983.

[9] V.A. Morozov, *Methods for Solving Incorrectly Posed Problems*, Springer-Verlag, New York, 1984

[10] T. Poggio and V. Torre, Ill-Posed Problems and Variational Principles in Vision. *MIT AI Memo 773*, 1984

[11] T. Poggio, H. Voorhees, and A. Yuille, A Regularized Solution to Edge Detection. *MIT AI Memo 833*, 1985

[12] C.H. Reinsch, Smoothing by Spline Functions, *Numer. Math*, Vol. 10, pp. 177–183, 1967

[13] D. Terzopoulos, Regularization of Inverse Visual Problems Involving Discontinuities. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, PAMI-8, No.4, July 1986

[14] A. Tikonov, Regularization of Incorrectly-Posed Problems. *Soviet Mathematical Journal*, vol-4, pp. 1624-1627, 1963

[15] A. Tikhonov and V. Arsenin, *Solution of Ill-Posed Problems*. Washington, DC: Winston and Wiley.

[16] V. Torre and T. Poggio, On Edge Detection. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, PAMI-8, No.2, pp. 147–163, March 1986.

[17] S. Weisberg. *Applied Linear Regression*. Wiley series in probability and mathematical statistics, John Wiley and Sons, 1980.

[18] A. Witkin, Scale-Space Filtering, *Proc. IJCAI*, pp. 1019–1021, Karlsruhe, Germany, 1983

[19] A.L. Yuille and T.A. Poggio, Scaling Theorems for Zero Crossings, *IEEE Transactions of Pattern Analysis and Machine Intelligence*, PAMI-8, No.1, pp. 15–25, Jan 1986.

[20] Y.T. Zhou and R. Chellappa. Linear feature extraction based on an AR model edge detector. In *ICASSP*, 1987.

[21] Y.T. Zhou, A. Rangarajan, and R. Chellappa. A Unified Approach for Filtering and Edge Detection in Noisy Images. In *Proceedings Twenty-first Annual Assilomar Conf. on Systems, Signals and Computers*, Pacific Grove, CA, November 1987.